BÖLÜM 2

2. VERİ ve VERİ MODELLERİ

2.1. Model Nedir?

Model kelimesi; isim, sıfat ve fiil olarak ve her birinde oldukça farklı çağrışımlar yapacak şekilde kullanılmaktadır. İsim olarak "model", bir temsili ifade eder. Bu temsil; bir mimarın, bir binanın küçük ölçekli modeli veya bir fizikçinin bir atomun büyük ölçekli modelini oluşturması anlamındadır. Sıfat olarak "model", mükemmeliyetin veya idealin ölçüsünü ifade eder. "Model ev", "model öğrenci" ve "model eş" ifadelerinde olduğu gibi. Fiil olarak "model" ise, bir şeyin nasıl olduğunu ispat etmek, açıklamak, göstermek anlamındadır.

Bilimsel modeller bütün bu çağrışımları bünyelerinde bulundururlar. Onlar; durumların, nesnelerin ve olayların temsilleridir. Gerçeklerden daha az karmaşık ve böylece araştırma amacıyla kullanılmaları daha kolay olduğundan, bu anlamda ideal hale getirilmişlerdir. Gerçek durumlarla karşılaştırıldıklarında, modellerin basitliğinin sebebi, gerçeklerin sadece uygun özelliklerini temsil etmelerinden kaynaklanmaktadır. Örneğin, yeryüzünün bir kısmının modeli olan bir yol haritasında, bitki örtüsü gösterilmez. Çünkü bu durum, o haritanın bir yol haritası olarak kullanımı açısından uygun değildir. Güneş sisteminin bir modelinde, gezegenleri temsil eden topların, gezegenlerle aynı maddeden yapılmış olmaları veya aynı sıcaklığa sahip olmalarına ihtiyaç yoktur.

Bilimsel modellerden, gerçeklerin farklı boyutları hakkındaki bilgiyi artırmak ve birbirleri ile ilişkilendirmek için faydalanılır. Modeller, gerçeği ortaya çıkarmak ve bundan daha fazla olarak, geçmişi ve şimdiki durumu açıklamak ve geleceği tahmin ve kontrol etmek için kullanılır. Modeller uygulanarak, gerçekler üzerinde bilimin kontrolü sağlanır. Modeller gerçeğin tarifi ve açıklamasıdır. Bir bilimsel model, aslında, gerçek hakkında bir veya bir dizi ifadelerdir. Bu ifadeler olaylara dayanan, kanun benzeri ya da teorik olabilir.

Bilimde, sıradan işlemlerde olduğu gibi, değişik tipte modeller kullanılır: Simgesel Model, Benzetim Modeli ve Sembolik Model.

Simgesel Modeller, durumların büyük veya küçük ölçekli temsilleridir. Gerçek şeylerin uygun özelliklerini temsil ederler. Şekilleri, temsil ettikleri şeylere benzer. Yol haritaları, hava fotoğrafları bu tip modellere örnek verilebilir.

Benzetim Modelleri, bazı durumlarda ise; haritada yükseltiler, yol genişlikleri gibi özellikleri belirtmek gerekebilir. O zaman, renkler ve kontur çizgileri gibi bir takım açıklayıcı özelliklere ihtiyaç duyulur. Bu tip modeller Benzetim Modelleri olarak isimlendirilir.

Sembolik Modellerde, temsil edilen şeylerin özellikleri sembollerle ifade edilir. Böylece, bir grafik ile gösterilen ilişki (benzetim modeli), bir eşitlik olarak da ifade edilebilir. Bu tip modellere Matematiksel Modeller de denilmektedir.

Bu üç tip modelden benzetim modeli, soyut ve geneldir. Matematiksel model ise en soyut ve en genel modeldir. Üzerinde düzenleme yapılabilmesi daha kolaydır. Simgesel modellerin ise anlaşılması diğerlerine göre daha kolaydır. Bilişim sistemlerinin oluşturulması için kullanılan veri modelleri, benzetim modelleri ve sembolik modellerdir.

Bir bilişim sisteminin kullanıcısı, özellikle bir karar verici, kendisini sonsuz denebilecek boyutta bilgi karşısında bulur. Bir bilişim sistemi modeli, gerçek bilgi kümesinin alt kümesini oluşturur ve onun daha basit bir şeklidir. Bu şekil, işlenebilmeye imkân verir ve bunu kullanarak elde edilen çözüm veya cevap, gerçek hayatta uygulanmaya çalışılır. Model, var olan bilgi yığınına bir düzen getirmeyi, hatta bir yapı oluşturmayı amaçlar. Tek bir model yoktur. Var olan bilgi yığınına, uygulanan farklı modeller doğal olarak farklı yorumlar getirir.

Gerçek hayattan alınan bir olayın modelinin iki tip özelliğinin olması gerekir. Birincisi, statik özellikler, ikincisi de dinamik özelliklerdir. Statik özellikler zamana göre değişiklik göstermez yada çok az gösterir. Dinamik özellikler ise bunun tam tersi olarak devamlı değişkendirler. Bu durumda, herhangi bir model (M), o modeli oluşturan kurallar kümesi (K) ve işlemler kümesinin (İ) bir fonksiyonu olarak tanımlanabilir.

$$M = f(K,İ)$$

Modeli oluşturan kurallar kümesi (K), veri modelinin statik özelliklerini temsil eder ve Veri Tanımlama Dili'ne (VTD) karşılık gelir. (M) veri modeli içinde, veri için izin verilen yapıların tanımlanması için kullanılır. Mümkün olan yapılar, birbirini tamamlayan iki şekilde belirlenir. Nesneler ve ilişkiler, kategorilerinin belirlenebilmesi için genel kurallar kullanılarak tespit edilir. Modelde bulunmasına izin verilmeyecek olan nesneler veya ilişkiler, sınırlar tespit edilerek hariç tutulur. Örneğin, bir işçi veri tabanında, her işçinin bir sigorta numarasının olması ve yöneticisinden fazla kazanmaması gibi sınırlar tespit edilebilir.

Gerçek hayattaki dinamik özelliklerin modelde kullanılabilmesini işlemler kümesi sağlar ve Veri Yönlendirme Dili'ne (VYD) karşılık gelir. Di gibi bir veri tabanı oluşumundan Dk gibi başka bir veri tabanı oluşumu elde etmek için yapılmasına izin verilen işlemleri tanımlar.

2.2. Veri Kavramı

Birinci bölümde veri kelimesi tanımlanmış ve yanlış verinin depolanmasını ve/veya verinin istenmeyen kişilerin kullanımına sunulmasını engelleyen bir takım imkânların olması gerektiği belirtilmişti.

Yanlış verinin iki türlü kaynağı olabilir: Programlama hataları, klavyeden hatalı giriş nedeniyle oluşan yanlışlıklar ve veri tabanı programının kötü niyetli kullanımı. Veri tabanlarının korunması iki başlık altında incelenebilir

1. Veri güvenliği,
2. Veri bütünlüğünün sağlanması.

2.2.1. Veri Güvenliği

Veri güvenliğinin konusu, veri tabanını, dolayısıyla veriyi yetkisiz kullanımlara karşı korumaktır. Bu konuda çok çeşitli yaklaşımlar vardır. Hem verinin istenmeyen şekilde değiştirilmesine veya zarar görmesine hem de yetkisiz kullanımlara engel olmak gerekir. Bunu sağlamak için bazı genel teknikler geliştirilmiştir.

• **Kullanıcıların tanımlanması:** Çok kullanıcılı ortamlarda farklı yetkilere sahip kullanıcılar vardır. Farklı yetkilere sahip kişilerin, veri tabanında ulaşabilecekleri veri farklıdır. Örneğin, bilgisayara veri girişi yapan bir işletimcinin, kurumun muhasebe kayıtlarına, muhasebe müdürü kadar yetkiliymiş gibi girerek değişiklikler yapması engellenmelidir. Bu amaçla, hangi kullanıcıların hangi yetkilerinin olduğu ve bu yetkilerini kullanabilmek için gerekli şifreler daha önceden tespit edilmelidir.

• **Fiziksel koruma:** Şifre sisteminin yeterli olmadığı durumlarda, verinin fiziksel koruma altına alınması gerekir. Yangın veya hırsızlığa karşı verinin yedeklenmesinin yapılması gibi.

• **Kullanıcı haklarının temin edilmesi:** Sistemde hangi kullanıcının hangi yetkilere ve haklara sahip olduğu ve neler yapabileceğinin önceden belirlenmiş olması gerekir. Bir kişinin yetkisini veya hakkını başka bir kişiye vermesi ise, sistemin müdahalesi dışında gerçekleşen bir durumdur.

Özellikle veri tabanının sorgulanmasında güvenlik problemleri ortaya çıkmaktadır. Hangi tür kullanıcının, hangi sorgu tiplerini sisteme yöneltebileceğinin daha önceden tespit edilmesi gerekmektedir. Fakat, yukarıda bahsedilen önlemlerden hiç biri tam bir koruma sağlamaz. Bu yüzden, birden fazla önlem kullanarak güvenlik artırılabilir.

2.2.2. Veri Tekrarı ve Veri Bütünlüğü

Bir veri tabanı yönetim sisteminde farklı veri dosyalarında; isim, adres, numara gibi bilgilerin bulunması gerekebilir. Örneğin, hem müşteri bilgilerini içeren bir veri tabanı dosyasında, hem de satılan malların sevkiyatının yapılacağı adreslerin bulunduğu başka bir veri dosyasında, müşteri adresi bilgilerinin yer alması gerekebilir. Yani, pek çok durumda, aynı verinin birden fazla veri dosyasında bulunması gerekebilir. Bu durum, veri tekrarı olarak ifade edilmektedir. Böyle bir durum,veri bütünlüğünün bozulmasına neden olur. Veri üzerinde yapılacak değişiklik, silme, ekleme gibi işlemlerin, o verinin bulunduğu bütün dosyalarda gerçekleştirilmesi gerekir. Özellikle çok kullanıcılı ortamlarda bu işlem oldukça önemlidir. Aksi taktirde, veri tabanında uygun olmayan veri ile çalışılmış olur. Veri bütünlüğünün bozulmasının bir sebebinin, veri tekrarı olduğu söylenebilir. Bir başka sebep de, verinin zayıf geçerlilik kontrolüdür. Bunun sebepleri de şu şekilde sıralanabilir:

• Veri güvenliğinin yetersiz oluşu,
• Veri tabanının zarar görmesi durumunda kurtarma yöntemlerinin yetersiz oluşu,
• Uzun kayıtların idaresinin zorluğu,
• Değişikliklerin esnek olmaması,
• Programlama ve bakım masraflarının yüksek olması,
• İnsandan kaynaklanan hatalar.

Günümüzde kullanılan çeşitli veri tabanı yönetim sistemi programları, yukarıda sayılan bütün problemlerin üstesinden hemen hemen gelebilecek çözümler üretmişler ve bunları kolay kullanılabilir hale getirmişlerdir. Kullanıcıların, bir veri tabanı oluştururken, ayrıca bu problemler için önlem almalarına gerek kalmamaktadır.

2.3. Veri Modeli

Bir veri modeli, verinin hangi kurallara göre yapılandırıldığını belirler. Fakat yapılar, verinin anlamı ve nasıl kullanılacakları hakkında tam bir açıklama vermezler. Veri üzerinde yapılmasına izin verilen işlemlerin belirlenmesi de gerekir. İşlemler, yapının sunduğu çerçeve içinde çalıştırılırlar.

2.3.1. Yapılar

Veriyi yapılandırma ve görüntüleme mekanizmalarından biri soyutlamadır. Soyutlama, detayları gizleme ve genel üzerinde yoğunlaşma yeteneğidir. Veri modellemesinde soyutlama, veri kategorilerini elde etmek için kullanılır.

Veri yapılarının oluşturulmasında kullanılan kavramlardan biri de kümelerdir. Bir küme, düzgün bir şekilde tanımlanmış ve bir üyelik koşulu tarafından temsil edilen nesneler topluluğudur. Üyeleri az ya da çok homojen olan kümeler vardır. Örneğin, 10 ile 20 arasındaki tam sayılar, uzunluğu 20 karaktere kadar olabilen alfanümerik değerler gibi. Bu homojen kümeler, tanım kümesi olarak isimlendirilirler. Semantik açıdan bir anlam taşıyan nesneyi temsil eden, isimlendirilmiş bir tanım kümesi (örneğin, MAAŞLAR), öznitelik olarak isimlendirilir.

Veri yapılarının unsurlarından biri de ilişkilerdir. İlişki, kümelerin toplanmasını ifade eder. Aynı zamanda kendisi de bir kümedir ve semantik olarak belirli bir karşılığı yoktur. Fakat, veri modellemesinde ilişki, iki nesne arasındaki ilişkiyi gösteren bir tip olarak tanımlanabilir. Örneğin, İŞÇİ ve İŞYERİ arasında bir İŞ ilişkisi vardır. Bir ilişkiye uygulanabilecek semantik bir tercüme, her satırı bir varlığa karşılık gelecek şekilde belirlemektir. Varlığın tam bir tanımı olmamasına rağmen, objektif bir gerçekliği olan veya olduğu düşünülen şey olarak tarif edilmektedir. Örneğin, İŞÇİ bir varlık tipi olarak belirlenebilir. Bu varlık tipinin özellikleri de, İŞVEREN, İSİM, ADRES, YAŞ, BÖLÜM, TECRÜBE ve MAAŞ olabilir.

Bir veri yapısı oluşturulurken, verinin bir şekilde bilgisayara yerleştirilmesi söz konusu olduğu için, nesneler ve onlar arasındaki ilişkilerin temsil edilmesi gerekir. Bu tür bir temsil tablolarla yapılabilir. Bir tabloda sütun başlıkları olarak öznitelikler ve satırlarda da bu özniteliklerin aldığı değerler (kayıt birimleri) yer alır. Tablodaki her bir sütun, bir veri birimidir.

Düz bir dosyadan oluşan veri tabanları olabileceği gibi (örneğin isim ve adres alanlarından oluşan adres veri tabanları), birden fazla dosyadan oluşan veri tabanları da vardır ve daha yaygın bir şekilde kullanılmaktadır. Bir veri tabanında temsil edilebilecek genel kayıt ilişkilendirme tipleri vardır. Bunlar şu şekilde sıralanabilir:

• **Bire bir ilişkiler (one-to-one relationships):** Aralarında bir ilişki olan iki tablo arasında, tablolardan birindeki asıl anahtar alanın kayıt değerinin, diğer tablodaki sadece bir kayıtta karşılığının olması durumunu gösteren ilişki tipi. **Örnek :** bir işçinin doğum yeri bilgisinin doğum yerleri tablosunda bir şehre karşılık gelmesi gibi.

• **Tekil çoklu ilişkiler (one-to-many relationships):** Aralarında bir ilişki olan iki tablo arasında, asıl anahtar alanın kayıt değerinin, diğer tablodaki birden fazla kayıtta karşılığının olması durumunu gösteren ilişki tipi. **Örnek :** Bir öğrencinin birden fazla almış olduğu derse ve bu derse ait vize final sınav sonuçları gibi. Bir öğrenciye karşılık birden fazla ders notu.

• **Çoğul tekli ilişkiler (many-to-one relationships):** Aralarında bir ilişki olan iki tablo arasında, tablolardan birindeki bir kaydın değerinin, asıl anahtar alanın olduğu diğer tabloda, birden fazla kayıtta karşılığının olması durumunu gösteren ilişki tipi.

• **Çoklu ilişkiler (many-to-many relationships):** Aralarında bir ilişki olan iki tablo arasında, tablolardan herhangi birindeki herhangi bir kaydın, diğer tablodaki birden fazla kayıt ile ilişkilendirilebildiği ilişki tipi.

2.3.2. Kısıtlar

Veri üzerindeki mantıksal sınırlamalara kısıt adı verilir. Kısıtların genel olması tercih edilen bir durumdur. Örneğin, "Tüm yöneticilerin maaşları, işçilerinden daha fazladır" ifadesi, "Ali Bey'in maaşı Veli Bey'in maaşından daha fazladır" ifadesinden daha geneldir ve dolayısıyla daha kullanışlıdır. Kısıtlar, veri modellerinde bütünlük sağlamak ve semantik nedenlerle kullanılır. Kümeler üzerinde kullanılabilir. Örneğin, "İŞÇİ varlık tipinin YAŞ özniteliği 15 ve 65 arasında değer alabilir" şeklinde bir kısıtlama veri modelinde uygulanabilir. Bu sayede gerçek dünyada karşılaşılan bir özellik, oluşturulacak veri tabanına yansıtılabilir.

Bir ilişki, iki veya daha fazla kümenin elemanları (nesneler) arasında mümkün olabilecek tüm kombinasyonları içerir. Bu işleme haritalandırma denir. Kısıtlamalar ilişkiler üzerinde belirlendiği zaman, nesneler arasındaki bir takım anlam ifade etmeyen, fakat teorik olarak mümkün olabilen ilişkilerin, gereksiz yere modele yerleştirilmeye çalışılması önlenmiş olur.

Tablolarda kısıtların kullanılması, fonksiyonel bağımlılıkların belirlenmesi amacını taşımaktadır. Örneğin, İŞÇİ varlık tipinde İŞÇİ_NO özniteliği bir aday anahtar olabilir. Çünkü, diğer öznitelikler, bu özniteliğe bağlı olabilir. Başka aday anahtarlar da olabilir (İSİM, ADRES gibi). Bir tabloda, aday özniteliklerinden biri asıl anahtar olarak belirlenir. İki ayrı ilişki tipi arasında bir bağlantı kurabilmek, o ilişkinin bir tablosundaki bir anahtarın, diğer tabloya eklenmesi ile mümkün olur. Buna türetme, ikinci tabloya eklenen anahtara da yabancı anahtar denir.

2.3.3. İşlemler

İşlemler, bir veri tabanı durumundan, bir başka veri tabanı durumu elde etmek için yapılan işlemlerdir. Bunlar, verinin çağırılması, güncellenmesi, eklenmesi veya silinmesi ile ilgili işlemlerdir. Çok kesin seçimler üzerinde yapılır. Bunların yanında, daha genel işlemler de vardır. Örneğin; bütünlük mekanizması, toplam fonksiyonları (istatistiksel fonksiyonlar da bunlar arasındadır), veriye ulaşım kontrolleri gibi. Bu mekanizmalara veri tabanı yöntemleri denir. Bu mekanizmalar, CODASYL tarafından yayınlanmıştır.

2.4. Başlıca Veri Modelleri

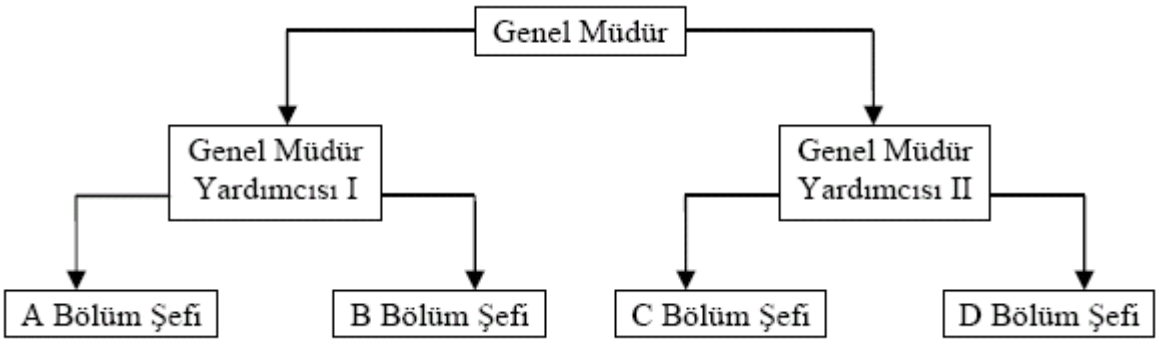
Veri modellemesi yapmak amacıyla pek çok veri modeli geliştirilmiştir. Fakat, bütün modeller aynı özellikleri taşımaz. Her modelin farklı durumlara uygun olan çeşitli özellikleri vardır. O yüzden, modeller arasında tam bir sıralama yapmak mümkün değildir. Bununla birlikte, yetersiz de olsa bir sınıflama yapılabilir.

2.4.1. Basit Veri Modelleri

Basit veri modelleri olarak ayrılan ilk grup veri modelleri, bilgisayarlarda veri işleme ihtiyacının ortaya çıkmasıyla, dosyalama sistemleri oluşturmak amacıyla kullanılmaya başlanan Hiyerarşik ve Şebeke veri modelleridir.

2.4.1.1. Hiyerarşik Veri Modelleri

Hiyerarşik veri modellerinde çoklu ilişkileri temsil edebilmek için, varlık tiplerinin her ilişki için ayrı ayrı tanımlanması gerekir. Bu da gereksiz veri tekrarına sebep olur. Hiyerarşik model, bir ağaç yapısına benzer. Model dahilindeki herhangi bir düğüm, altındaki n sayıda düğüme bağlanırken, kendisinin üstünde ancak bir düğüme bağlanabilir. Hiyerarşik yapının en tepesindeki düğüm noktasına kök denir ve bu düğümün sadece bağımlı düğümleri bulunur. Bu veri yapısını gösteren grafiğe de hiyerarşik tanım ağacı denir.

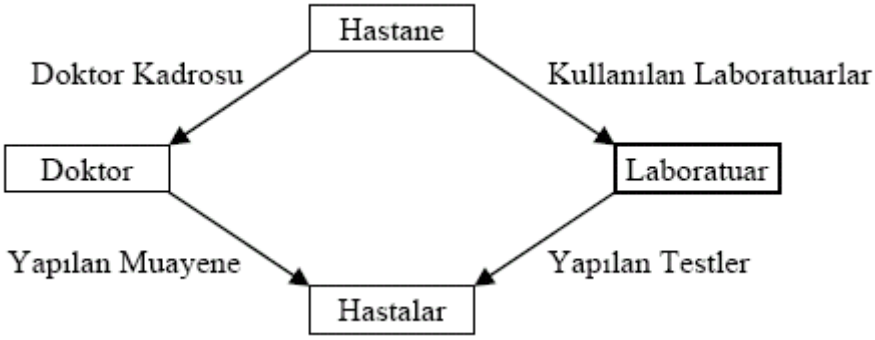


Şekil 2.4.1.1.1. Hiyerarşik Tanım Ağacı

2.4.1.2. Şebeke Veri Modelleri

Şebeke veri modelleri, tablo ve grafik temellidir. Grafikteki düğümler varlık tiplerine karşılık gelir ve tablolar şeklinde temsil edilir. Grafiğin okları, ilişkileri temsil eder ve tabloda bağlantılar olarak temsil edilir. Spesifikasyonu, 1971 yılında DBTG-CODASYL tarafından belirlenmiştir.

İki ayrı veri yapılandırma aracı vardır: Kayıt tipi ve bağlantı. Kayıt tipleri varlık tiplerini belirler. Bağlantılar ise, ilişki tiplerini belirler. Bu yapıyı gösteren grafiğe de veri yapısı grafiği adı verilir.



Şekil 2.4.1.2.1. Şebeke Veri Yapısı Grafiği

Şebeke veri modeli, veri modelleri içinde en genel olanlarından biridir. Şebeke içinde bir eleman, herhangi bir başka elemana bağlanabilir. Hiyerarşik yapılardan farklı olarak, şebeke yapılarında bağlantı açısından herhangi bir sınırlama yoktur. Şebeke veri modelleri, düğümler arasında çoklu ilişkiler kurulamadığı için, kısıtlı bir veri modeli olarak kabul edilir. Hiyerarşik veri modelleri ise, daha da kısıtlı bir veri modelidir. Şebeke veri modelinde kullanılan işlemler, ilişkisel veri modelinde kullanılan işlemlerin benzeridir. Fakat, şebeke veri modellerinde bağlantılar tarafından belirlenmiş ilişkiler dışında, kayıt tipleri arasında ilişki belirlenemez.

2.4.2. Geliştirilmiş Veri Modelleri

1960 ve 1970'li yıllarda hiyerarşik veri modeli üzerine geliştirilmiş veri tabanı yönetim sistemleri ile, daha sonra, şebeke veri modeli ile çalışan VTYS yaygın kullanımda iken, teorik temelleri ve deneysel uygulama ve geliştirme aşamaları, 1970'li yıllarda tamamlanmış olan ilişkisel veri modeline dayalı VTYS, 1980'li yıllarda ticari kullanıma girerek çok hızla yaygınlaşmışlardır.

Geliştirilmiş veri modelleri, Varlık-İlişki Veri Modelleri, İlişkisel Veri Modelleri ve Nesne Yönelimli Veri Modelleri olarak sıralanabilir.

2.4.2.1. Varlık-İlişki Veri Modelleri (Vİ Modeli)

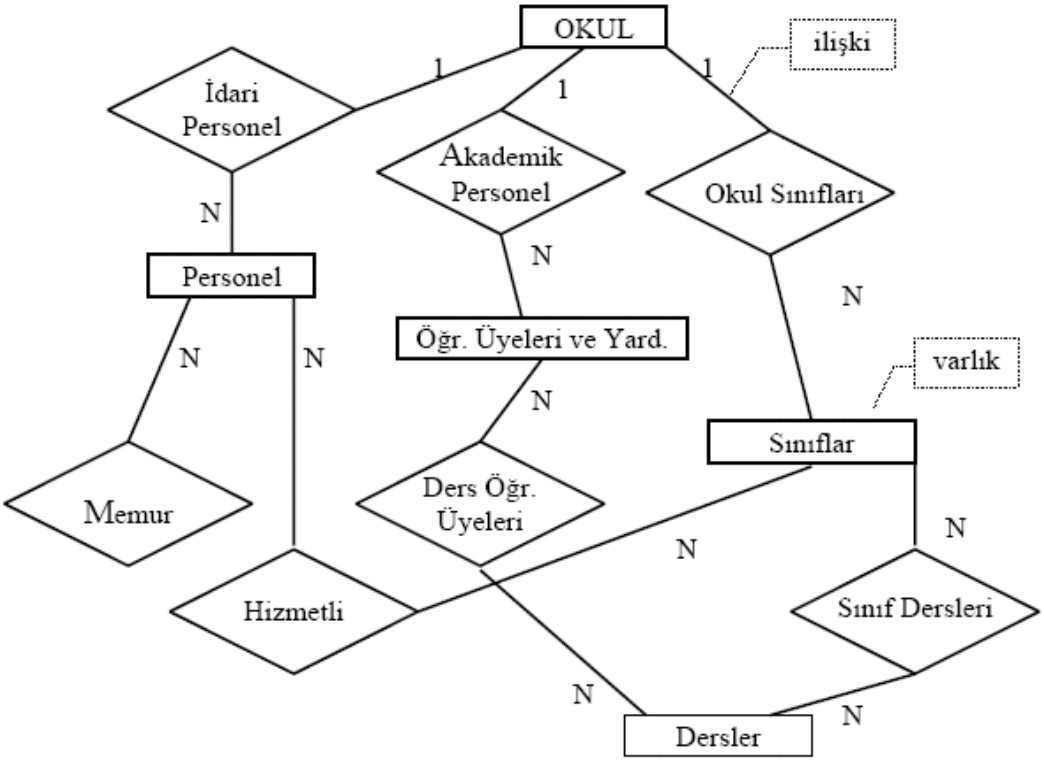
Bir veri tabanı uygulaması için varlık, hakkında tanımlayıcı bilgi saklanabilen herşey olarak kabul edilmektedir. Varlık, bağımsızdır ve tek başına tanımlanabilir. Bir varlık, ev, öğrenci, araba gibi bir nesne ya da futbol maçı, tatil, satış gibi olaylar olabilir. En anlamlı şekilde kendi öznitelikleri tarafından temsil edilir. Örneğin, bir EV; öznitelikleri olan ADRES, STİL, RENK ve MALZEME ile tanımlanabilir. Eğer bir özniteliğin kendisi tanımlayıcı bilgi içeriyorsa, onu varlık olarak tanımlamak gerekir. Örneğin, eğer evin malzemesi hakkında ek bilgi depolamak gerekiyorsa MALZEME'yi de varlık olarak sınıflamak gerekir.

Varlık-İlişki veri modelleri (Vİ), sütunlarında, öznitelikleri temsil eden değişkenlerin yer aldığı ve satırlarında da enstantanelerin temsil edildiği tablolar, varlıklar ve aralarındaki ilişkileri oklarla göstermek için kullanılan grafikler üzerine kurulmuş veri modelleridir.

Ticari veri tabanlarında yaygın olarak kullanılan veri modellerinden biridir. Şebeke ve hiyerarşik veri modelleri ile ortak noktaları vardır. Fakat, veri tabanı tasarım süreçleri için kullanılmak maksadıyla geliştirildiklerinden bu iki modelin genelleştirilmiş şeklidir. Çoklu ilişki tiplerinin doğrudan modelde kullanılmasına izin verir. Bu modelde, kurum şeması kavramı söz konusudur. Bu şema, kurumun tüm verisinin görünümünü temsil eder ve fiziksel sınırlamalardan bağımsızdır. Aynı zamanda, bu şema ANSI/X3/SPARC kavramsal şemasına çok benzemektedir. Aralarındaki temel fark, kavramsal şemanın, dahili şema ve harici şema arasında haritalandırma yapabilmesidir. Temelde, Vİ veri modeli, veri tabanının mantıksal özelliklerinin bir dokümantasyonudur. Vİ modeline göre düzenlenen veri tabanının yapısı, ***Varlık-İlişki Diyagramı*** ile gösterilir.

Şebeke ve hiyerarşik veri modellerinde, sadece ikili fonksiyonel bağlantılara izin verilmektedir. Vİ veri modelinde ise, varlıklar arasında n adet ilişki tanımlanabilir. Bu ilişkiler, bire bir, fonksiyonel veya çoklu olabilir. Tekrar eden bağlantılar da kullanılabilir.

Öznitelik, varlık veya ilişki ile bunların aldığı değerler arasındaki haritalandırmayı temsil eder. Bazı özniteliklerin birden fazla değeri olabilir. Örneğin, telefon numarasını bir öznitelik olarak kabul edersek, bir şirketin birden fazla numarası olabilir. Fakat, doğum günü özniteliği ele alındığında, her bir kişinin bir doğum günü olduğundan, bu öznitelik, çok değere sahip değildir.



Şekil 2.4.2.1.1. Varlık – İlişki Diyagramı

Vİ modeli ilk olarak ortaya konulduğunda (1976) bir veri dili geliştirilmemişti. Bunun anlamı, bilgi sorgulamalarının küme işlemleri ile yapılması demekti. Daha sonra, veri modeli için CABLE (ChAin-Based LanguagE) dili geliştirildi. Vİ modellerinin en büyük avantajlarından biri, uzman olmayan kişiler tarafından da anlaşılabilecek yapıda olmasıdır. Üzerinde düzeltme işlemleri kolayca yapılabilir. Bu açıdan belirli bir veri tabanı yönetim sistemine bağlı değildir.

2.4.2.2. İlişkisel Veri Modelleri

İlişkiler ve onların temsilleri olan tablolardan oluşan veri modelleri ilk olarak 1970 yılında Codd tarafından ortaya atılmıştır. İlişkisel veri modelleri formüle edilirken, veri yönetimi ihtiyaçlarını karşılayabilmek için ilişkinin matematiksel teorisi, mantıksal olarak genişletilmiştir. İlişkisel veri modellerinde kullanılan tek yapılandırma aracı ilişkidir. İlişkinin tanımı, veri tabanı ilişkilerinin zamana bağlı olması dışında, matematiksel tanımı ile aynıdır. Yani, bir veri tabanı ilişkisinde satırlar, eklenebilir, değiştirilebilir yada düzeltilebilir. Aşağıdaki örneklerde büyük harflerle yazılan ifadeler ilişki isimlerini, parantez içindeki ifadeler de tanım kümesi isimlerini göstermektedir.

HASTANE(Hastane_Kodu, Hastane_Adı, Adres, Tel_No, Yatak_Sayısı)

DOKTOR(Hastane_Kodu, Diploma_No, Adı, Uzmanlığı)

İŞÇİ(Sigorta_No, Adı, Adres, Kıdem, Maaş, Yaşı)

Şekil 2.4.2.2.1. Tanım Kümesi

Yukarıdaki satırlar, basit bir hastane veri tabanının ilişkisel şemasını göstermektedir. İlişkisel şema, ilişki isimlerinin ve karşılık gelen tanım kümesi isimlerinin listesidir. Varlık tiplerini belirlemekte kullanılır.

HASTANE

Hastane_Kodu	Hastane_Adı	Adres	Tel_No	Yatak_Sayısı
1	Marmara Üniv.	Altunizade	216 333 33 33	150
2	HP Numune	Haydarpaşa	216 333 33 34	150
3	Kartal Devlet	Kartal	216 333 33 35	120
4	Haseki	Fındıkzade	212 555 55 55	100

Şekil 2.4.2.2.2. İlişkisel Tablo

İlişkisel şema listesini oluşturan her bir satır, bir tablo olarak temsil edilir. Tablonun sütunları öznitelik olarak isimlendirilir. Örneğin, HASTANE tablosunun öznitelikleri; Hastane_Kodu, Hastane_Adı, Adres, Tel_No ve Yatak_Sayısı'dır.

Tablonun satırlarında bütün özniteliklerin aynı değerler aldığı iki satır olamaz. Her satır diğerinden mutlaka farklıdır. Aksi halde veri tekrarı söz konusu olur. Veri tabanlarındaki ilişki kavramı, matematikteki küme kavramını esas aldığı için aynı satırın bir tabloda birden fazla yer alması mümkün değildir. Bu nedenle, ilişki için bir anahtar kullanmak gerekir. Anahtar, bir satırı tek başına tanımlayabilen öznitelikler kümesidir. Anahtar kavramı, ilişkisel veri modelinde kullanılan önemli bir kısıttır.

Bu kurallar kullanılarak hazırlanan bir ilişkisel modelde, yine de belirsizlikler ve uyumsuzluklar bulunabilir. Bunları gidermek için de bir dizi düzgüleme işlemine gerek duyulabilir. Düzgülemek, veri tabanı tasarım prensiplerini yapısallaştırmayı amaçlar. İlişkiler ve öznitelikler arasındaki fonksiyonel bağımlılıkları düzenler. Birbirini takip eden beş işlemden oluşur. Fonksiyonel bağımlılık şu şekilde tarif edilebilir: "x ve y öznitelikleri arasındaki ilişki R ile gösterildiğinde, her bir x değerine bir tek y değeri karşılık geliyorsa, R'nin y özniteliğinin, R'nin x özniteliğine fonksiyonel olarak bağımlı olduğu söylenir."

Veri üzerinde yapılacak işlemler için, ilişkisel veri modellerinde üç tip dil kullanılır. Birincisi, matematikteki ilişkisel işlemlere dayanır. Bu tip dillere örnek olarak INGRES ve QUEL verilebilir.

İkinci tip dil, görüntü yönelimlidir. Boşluk doldurma yöntemiyle çalışır. Örneğin, QBE (Query By Example) ve CUPID bu tür dillerdendir.

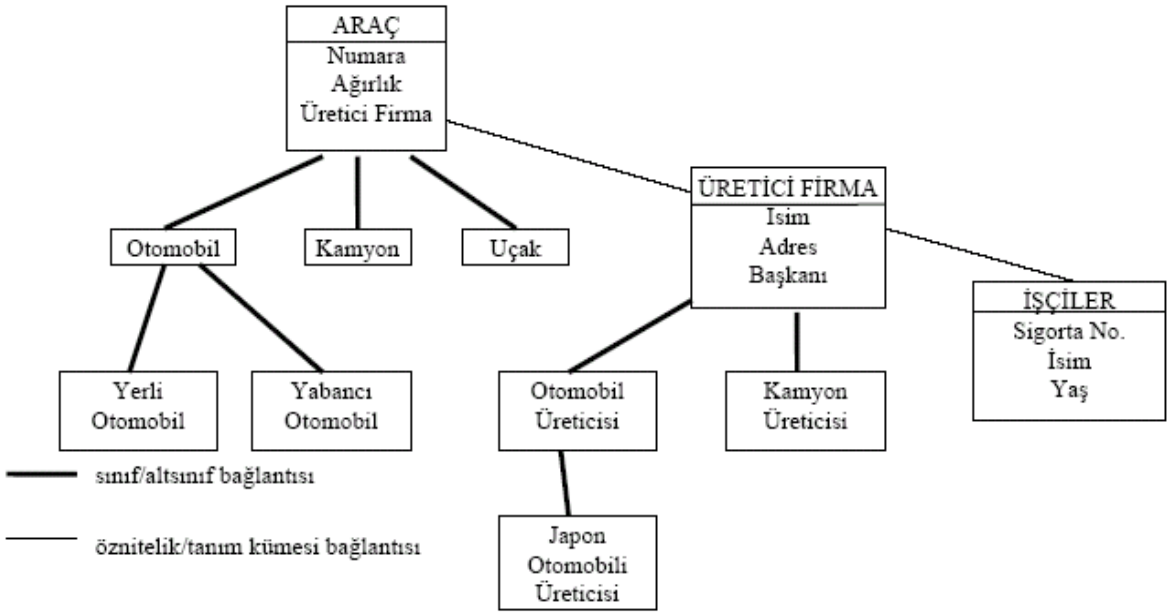
Üçüncü tip dil, haritalandırma yönelimli dildir. Bu tip diller, bilinen bir özniteliğin ya da öznitelik kümesinin, aranan bir özniteliğin ya da öznitelik kümesinin üzerinde, bir ilişki yoluyla haritalandırılması prensibiyle çalışır. Örneğin, yapısal sorgulama dili (SQL) bu tip bir veri dilidir. SQL ilerleyen bölümlerde detayları ile anlatılacaktır

2.4.2.3. Nesne Yönelimli Veri Modelleri

Nesne yönelimli sistemler, bir istatistiksel sistem içinde, esnek veri yapılarının geliştirilmesi ve istatistiksel modellerin sunumunda da kullanılmaktadır. Nesne yönelimli programlamanın başlangıcı, 1960'ların sonu ve 1970'lerin başı arasında geliştirilen simülasyon dili Simula'ya kadar uzanır.

Nesne yönelimli veri modelinde, bir sorgunun karşılığında mutlaka önceden tanımlanmış belirli bir nesne kümesi olması gerekir. Bir sorgunun sonucu olarak tesadüfi bir nesne kümesinin elde edilmesi mümkün değildir. Çünkü bütün nesnelerin, modelde önceden tanımlanmış olması gerekmektedir. İlişkisel modeldeki ilişki kavramı, nesne yönelimli modelde sınıf kavramına karşılık gelmektedir.

Nesne yönelimli modellemenin en önemli faydalarından bir tanesi de, modeldeki nesneleri tanımlarken, ortak öznitelik ve metotlara sahip nesnelerin kullanıldıkları her farklı ortamda, tekrar tanımlanmalarına gerek duyulmamasıdır. Aksi taktirde bu durum, hem tekrardan dolayı yer kaybına, hem de modeldeki dinamik değişikliklerin pratik olmamasına sebep olacaktır. Nesne yönelimli veri modelindeki sınıf hiyerarşisi ve kalıtım özelliği, bu olumsuz durumu ortadan kaldırarak, nesnelerin özniteliklerinin ve metotlarının yeniden kullanımına imkân vermektedir. Çünkü bir sınıf, ait olduğu üst sınıfın tüm özelliklerini taşır ve o sınıftaki nesneler, modelin başka bir yerinde kullanılacağı zaman yeniden tanımlanmaya gerek kalmadan tekrar kullanılabilir.



Şekil 2.4.2.3.1. Sınıf Hiyerarşisi

Genellikle soyutlama olarak anılan bu tip işlemler, şu başlıklar altında toplanabilir:

• **Sınıflandırma ve elemanlarına ayırma:** Sınıflandırma, nesne yönelimli veri modeli yaklaşımının temelini oluşturmaktadır ve aynı özellik ve davranışlara sahip nesnelerin nesne sınıfları içinde gruplanması ile ilgilidir. Bir sınıftaki nesneler, o sınıfın tanımına göre tarif edilebilir. Böylece her nesneyi ayrı ayrı tarif etmeye gerek kalmaz. Elemanlarına ayırma ise sınıflandırma işleminin tersidir ve bir sınıf içinde farklı nesneler oluşturulması ile ilgilidir. Aşağıdaki nesne yönelimli veri modeli buna bir örnektir.

```
CLASS Otel
    PROPERTIES
        isim : string;
        adres : string;
        sahibi : kurum;
        yönetci : kişi;
        servis : (yüzme_havuzu, sauna, tenis, bar, lokanta...)
        ...
    OPERATIONS
        Create (işemler)
        Rezervasyon (oda_no: integer; müşteri: kişi; geliş_tar, ayrılış_tar: Date
Type)
        ...
END Otel
```

Bu örnekteki sahibi, adres, servis gibi nesneler, Otel sınıfının elemanlarıdır ve tanımları da birbirinden farklıdır. Örneğin, Otel sınıfının bir elemanı,

İsim : İstanbul Oteli
Adres : Beşiktaş, İstanbul
Sahibi : (kurum elemanı)
Yönetici : (yönetici elemanı)
Servisler : yüzme_havuzu, sauna, tenis, bar, lokanta

şeklinde tanımlanırken, kurum nesnesinin bir elemanı, aşağıdaki gibi,

İsim : İstanbul Otelcilik A.Ş.
İdari Yeri : İstanbul
Telefon : 0212-555 5555

yönetici nesnesinin bir elemanı da, aşağıdaki gibi tanımlanabilir.

İsim : Ahmet Gel
Adres : Fenerbahçe, İstanbul
Doğum_Tar : 1943

• **Tanımlama:** Bu işlem hem soyut kavramların (sınıf), hem de somut kavramların (elemanlar), teker teker tanımlanması ile ilgilidir ve anahtar değerler yardımıyla yapılır.

• **Toplam:** Nesneler arasındaki ilişkilerin daha üst düzeyde, bir toplam nesne (veya tip) tarafından temsil edilmesi ile ilgili bir soyutlama yöntemidir. Bu toplam tipe genellikle anlamlı bir isim verilir ve bu isim modelin başka yerlerinde, ona ait özellikler referans olarak verilmeden kullanılabilir.

• **Genelleştirme:** Aynı özelliklere sahip bir grup nesnenin, soysal nesne olarak temsil edilmesi ile ilgili bir soyutlama yöntemidir. Örneğin, bir kurumda çalışan personel şu şekilde düzenlenebilir:

Bilgisayar Ekibi (Analizci, Programcı, İşletimci)
Çalışanlar (Bakım Ekibi, Bilgisayar Ekibi, Yönetici)

"Bilgisayar Ekibi" nesnesi; Analizci, Programcı ve İşletimci nesneleri için bir soysal nesnedir. Aynı şekilde Çalışanlar nesnesi de Bakım Ekibi, Bilgisayar Ekibi ve Yönetici nesneleri için soysal nesne durumundadır.

Nesne yönelimli veri modellerinin, ilişkisel veri modellerine karşı üstünlükleri vardır. Bunlar; NYVM'nde veri tiplerinin (tamsayı, gerçek sayı, alfanümerik değer, tarih vb.) İLVM'e göre daha esnek olması, nesne tanımlarında soyutlama yapılabilmesine imkân vermesi ve bu tanımların semantik içeriklerinin de olması sayesinde, veri bütünlüğünün daha kolay sağlanabilmesi ve ilişkisel veri modellerine göre, mevcut veri yapısında daha fazla genişleme ve yeniden düzenleme imkânlarına sahip olması sayılabilir.

Not : Bu bölüm
http://iletisim.marmara.edu.tr/bilisim/veri%20modelleri(csutcu%20dr%20tezinin%20bir%20bolumu).pdf
adresindeki dosyadan alınmıştır.